Query, maschere e VBA in Microsoft Access - Parte II

di Gianni Giaccaglini

SOMMARIO

(cliccare su uno dei titoli seguenti per andare al paragrafo corrispondente)

Riepilogando... 1
Le maschere di Access, in qualche dettaglio 2
Ingrandire, dimensionare & affini 3
Maschera singola, maschere continue e foglio dati 3
Le viste - anzi layout! - tabulare e giustificata 4
Altre particolarità di maschere e controlli 5
I pulsanti di spostamento 5
Interruzione di pagina e Struttura a schede 5
Il controllo Gruppo di opzioni 5
Modificabilità dei dati 6
Maschera di input con cautele VBA 8
Pulsanti con routine d'evento Clic ottenute con creazione guidata 8
Pulsanti con routine d'evento Clic senza creazione guidata 11
Routine d'evento tramite funzione su modulo standard 13
Maschere, sottomaschere e campi di totale 14
Creazione guidata di una maschera con sottomaschera 15
Creazione manuale di una maschera con sottomaschera 17
Caratteristiche delle maschere con sottomaschere 18
Risposta al quesito 19
Introduzione alle funzioni di totalizzazione 19

Riepilogando...

Siamo alla seconda puntata di una miniserie dedicata a chi, con Access, non è alle prime armi ma ancora non se la cava proprio bene con questo amichevole ma complesso programma. Nella prima parte abbiamo cercato di evidenziare le cose principali, ma prima di proseguire conviene fare il punto. Abbiamo cercato di evidenziare le necessarie distinzioni tra i diversi componenti di un database Access, parlando di tabelle, query e maschere. Per una scelta criticabile, abbiamo invece trascurato i report, che invece taluni autori mettono in cima alla loro trattazione. A buon diritto, visto che il progetto di un database dovrebbe partire fissando gli scopi che si vogliono raggiungere, e questi si estrinsecano il più delle volte con dei rapporti (report), destinati o meno alla stampa: rendiconti, totalizzazioni a più livelli, fatture, bolle e quant'altro. Ma questo avrebbe implicato mettere troppa carne al fuoco, tanto più che questo non è un corso organico di *progettazione* di un database, ma solo una raccolta di articoli. E così abbiamo deciso di rinviare il tema dei report, anche perché non vogliamo rendere troppo tardivo quello che

principalmente sta a cuore ai più esperti, ossia il **VBA** (Visual Basic per applicazioni) di Access. Come ripetiamo, un interlocutore privilegiato è da noi considerato chi proviene dal mondo Excel, con una certa pratica del relativo VBA.

Anche stavolta faremo riferimento, per le prove che ciascuno è invitato a fare personalmente, al database per gestione contatti, ottenuto con l'apposita creazione guidata: 1. **File ⇒ Nuovo...**; 2. scegliere **Gestione contatti**; 3. riempire con dati opportuni la struttura di tabelle, maschere e report che Excel crea, chiamata per default **Gestione contatti1**.

CONSIGLIO - Nei primi tempi può dar fastidio l'intervento della maschera **Pannello comandi** ad ogni caricamento di **Gestione contatti1.mdb**. Per evitarlo: 1. **Strumenti ⇒ Avvio...**; 2. nella finestra che segue sostituire con "(nessuna)" la voce "Pannello comandi" nella casella a discesa *Visualizza maschera/pagina.*

Ma vediamo di riassumere le nozioni e i concetti esposti nel precedente articolo (cose, che in diversi casi, riprenderemo), approfittandone per qualche precisazione.

- ***Access lavora su disco***. Chi proviene da Excel o Word, programmi che operano su RAM stenta, se non a comprendere questa diversità, a "farla propria", psicologicamente. In particolare non va mai dimenticato che, salvo nei casi di record protetti o di sola lettura, la più semplice modifica a un campo di una tabella o maschera, viene immediatamente registrata sul disco (*). D'altronde in Access non esiste un comando File ⇒ Salva relativo all'intero database (file .mdb). *Il salvataggio è però necessario quando si progetta un componente, quali tabella, query, maschera, report*. La cosa è ovvia con le query, per le quali è importante la distinzione fra quelle salvate (con un nome) o meno, ma vale per tutti i componenti, che possiamo, oltre che creare, modificare, attivando la visualizzazione **Struttura** (icona riga + squadra + matita). Ad esempio se apriamo una maschera e diamo un clic sull'icona appena detta possiamo non solo cambiarne le proprietà ma anche verificare l'effetto di tali modifiche passando nella vista Maschera. Se poi chiudiamo la maschera, in qualsiasi tipo di vista, Access ci invita al salvataggio.

NOTA (*) - Un dato consolatorio: se il programma va in crash non ci si deve preoccupare per la perdita di dati...

- ***Query*** e ***relazioni***. Non si faccia confusione (purtroppo è facile...). In entrambi i casi possono essere in gioco legami di *join* tra tabelle correlate, ma una query può anche interessare una singola tabella oppure basarsi su join differenti da quelle delle relazioni in senso stretto. Queste vengono fissate a livello database, ossia in sede di progetto, ed hanno lo scopo di garantire l'integrità referenziale. Su questo punto non possiamo che invitare a rileggere l'articolo precedente o, meglio ancora, un manuale organico, quel che ci preme sottolineare è che una query può fissare una join "estemporanea" diversa da quelle fissate nelle relazioni ma senza contraddire queste ultime.
- ***Campi calcolati e query associate***. Anche qui confondersi non è infrequente (specie se passa un po' di tempo tra una visita e l'altra di Access), soprattutto nel caso di controlli. A chi non è capitato di esitare sul da farsi - formula o query? - circa la proprietà *Origine controllo* di un campo? Il mondo di Access è popolato di una fauna composta da molte specie o, meglio, oggetti. All'impianto di fondo delle tabelle e delle relazioni fissate fin dal progetto di un certo database, si uniscono query, maschere, report per citare i più importanti). Questi componenti possono essere collegati e arricchiti in vari modi dai seguenti strumenti, che dovranno costituire un cocktail organico in un progetto ben strutturato:
  - ***clausole SQL***
  - ***dati calcolati***
  - ***routine Visual Basic*** (per applicazioni, ***VBA***).

Inoltre restano in vita, per compatibilità con precedenti versioni, le ***macro*** tipiche di Access.

Le maschere di Access, in qualche dettaglio

Chi proviene da Excel si rende subito conto della maggior complessità delle Form di Access. Il fatto più importante è che esse possono essere associate a tabelle o query, ma in prima battuta colpisce la loro maggior ricchezza, a partire dalla possibilità di esser dotate in alto a sinistra, per default, da tutti e tre i pulsanti di una finestra doc, cioè anche quelli di riduzione a icona e ingrandimento e non solo di chiusura. Questa e altre caratteristiche rendono molto affollata la finestra delle proprietà, al punto di rendere impossibile un'elencazione completa: persino nei testi più voluminosi molte cose vengono esposte cammin facendo.

Ingrandire, dimensionare & affini

Per offrire, in questo paragrafo, alcuni spunti anche spiccioli cominciamo con l'esaminare le proprietà legate ai tre pulsantini in alto a destra, riportando i nomi così come figurano nella finestra delle Proprietà della maschera (da attivare, nella vista Struttura, con un clic sull'icona manina-con-indice):

- ***Pulsanti ingrand./riduzione***, con le opzioni "Entrambi" (default); "Riduzione a icona consentita"; "Ingrandimento consentito" e "Nessuno";
- ***Pulsante chiusura***, che permette di scegliere tra "Sì" e "No".

Come è facile capire e provare, con la prima proprietà impostata a "Nessuno" e la seconda a "No" la finestra non può essere né ingrandita, né ridotta a icona, né chiusa (*)

NOTA (*) - Ma l'utente sveglio può chiudere la finestra con **Alt**+**F4**...

Già che ci siamo parliamo di altre proprietà affini della finestra (le opzioni relative sono indicate tra parentesi):

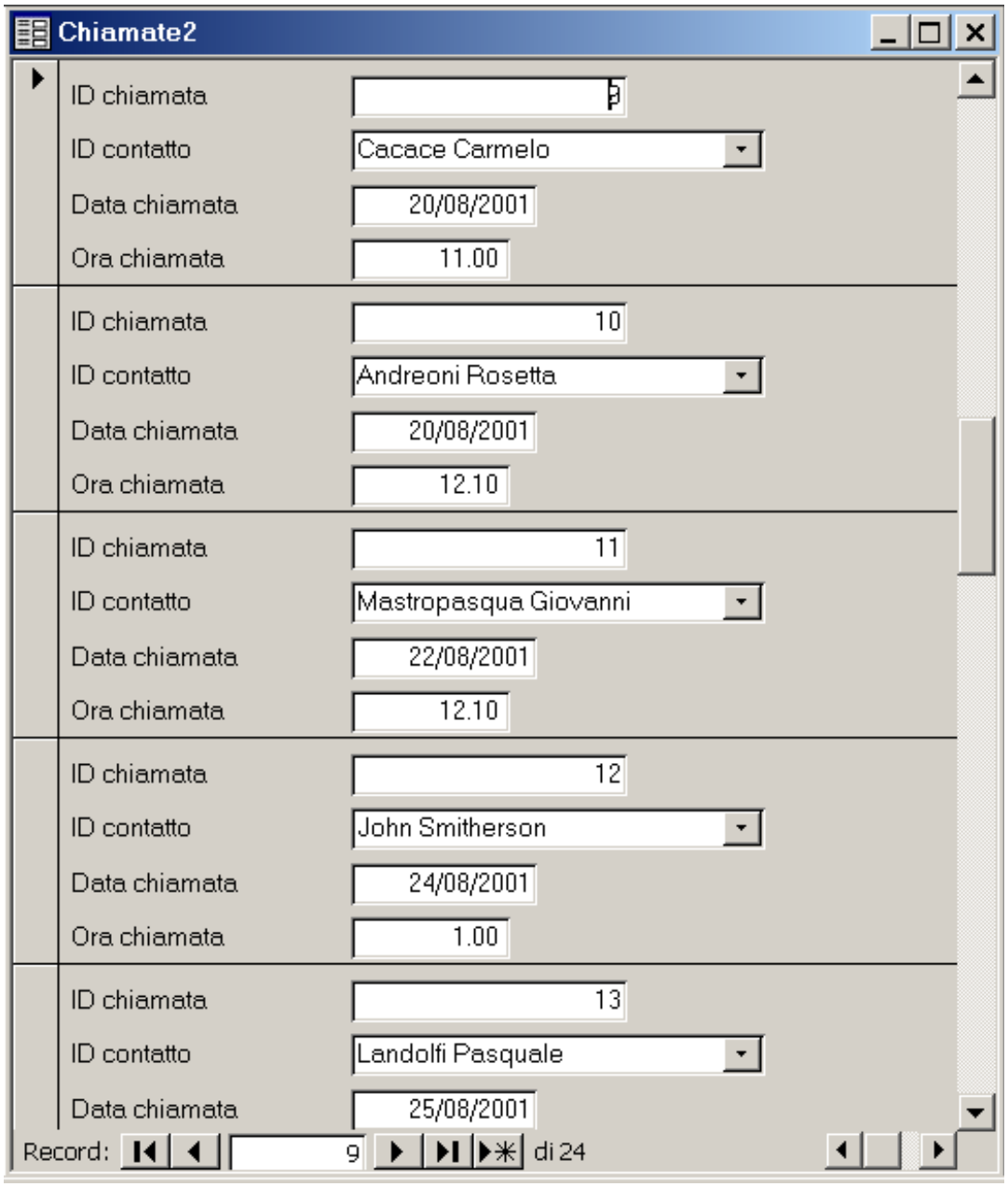
- ***Stile bordo*** ("Dimensionabile"; "Nessuna; "Dialogo"; "Sottile");
- ***Popup*** ("Sì"; "No");
- ***A scelta obbligatoria*** ("Sì"; "No").

Solo con *Stile bordo* = "Dimensionabile" si possono variare la larghezza e l'altezza della finestra, mentre "Dialogo" fa sparire i pulsanti di ingrandimento e riduzione (l'aspetto diventa quello della UserForm di Excel). Infine *Popup = "Sì"* fa sì che la maschera resti in primo piano, sovrastando le altre eventualmente aperte e *A scelta obbligatoria = "Sì"* impone all'utente una certa azione (tipicamente, clic su un pulsante che a sua volta lancia una macro).

Come si capisce, *Stile Bordo = "Dialogo"* e le altre due proprietà pari a "Sì", sono impostazioni tipiche per finestre di dialogo con l'utente.

Maschera singola, maschere continue e foglio dati

Piuttosto interessante è la proprietà ***Visualizzazione predefinita***, che accanto al default "Maschera singola" offre anche "Maschere continue" e "Foglio dati", nonché "tabella pivot" e "grafico pivot". Le ultime due meritano quasi un articolo a sé. Parlando più prosaicamente della seconda e terza, "Maschere continue" si presenta come illustrato nella figura qui sotto:



Come si vede la maschera presenta più di un record nella finestra, il che ha qualche utilità nel caso di maschere singole poco estese. Quanto a "Foglio dati", la vista è la stessa delle tabelle o query, per cui ci si può chiedere a che pro progettare una maschera... In effetti questa vista è di uso assai raro, *con l'importante eccezione delle sottomaschere*, ove spesso torna invece utile.

CONSIGLIO - Il lettore può sbizzarrirsi con le opzioni illustrate, o altre di suo gusto, *senza* sconvolgere maschere già progettate. Come? Alternando la vista Struttura con la visualizzazione Maschera e, alla fine degli esperimenti chiudendo la maschera e rifiutando l'invito di Access a salvala.

Di regola, comunque, si può sempre passare dalla visualizzazione maschera a quella foglio dati, ma esistono 4 proprietà *Consenti visualizzazione...* relative a: Foglio dati, Tabella pivot, Grafico pivot e Maschera, impostabili a "Sì" o "No". Si provi a fissare *Consenti visualizzazione maschera = "No"* e la nostra bella ed elaborata mascherina esibisce solo un arido elenco di record!

Le viste - anzi layout! - tabulare e giustificata

Su queste speciali possibilità di visualizzazione di una maschera regna, occorre ammetterlo, la confusione. A nostro avviso sono da evitare, ma non se ne può ignorare l'esistenza. Per capirci, facciamo un passo indietro (come nei romanzi ottocenteschi) invitando il lettore a seguire con cura la creazione guidata di una maschera, un tema che a dire il vero avevamo lasciato per scontato, ma che comunque riprenderemo in questo stesso articolo più avanti, a proposito di sottomaschere.

Ebbene a metà del percorso il wizard presenta un quadro in cui offre le opzioni **A colonne**; **Tabulare**; **Foglio dati**; **Giustificato**; **Tabella pivot** e **Grafico pivot**. Va subito chiarito un punto:

come indica il titolo del quadro, non si tratta di viste bensì di layout, ovvero stili di disposizione dei campi nella maschera.

Il layout a colonne è quello normale, con i campi disposti in verticale e le relative etichette sulla sinistra mantre il layout tabulare rassomiglia a quello foglio dati, salvo che i campi sono riportati in caselle separate, sormontate da più estetiche intestazioni fisse.

La confusione che si rischia (scagli la prima pietra chi non ci si imbatte prima o poi) è tra la *Visualizzazione* continua e il *layout* tabulare. Ma è una confusione che non va fatta.

Per gli altri layout si rimanda a personali esperimenti o a future trattazioni.

Altre particolarità di maschere e controlli

Molto lungo sarebbe il discorso sui dettagli relativi alle maschere e ai controlli in esse incorporati. Lo spazio tiranneggia e poi si finirebbe per annoiare, pertanto qui accenniamo alle cose principali, rinviando la trattazione ai momenti in cui si rendesse necessaria. Tralasciamo gli orpelli estetici relativi ai controlli, accennando solo agli sfondi, precostituiti o con immagini, per le maschere: nei due casi si danno i comandi **Formato ⇒ Formattazione automatica...** e, rispettivamente, **Inserisci ⇒ Immagine**..., facili da usare (in visualizzazione Struttura).

I pulsanti di spostamento

Non c'è quasi bisogno di dire cosa sono queste freccette di cui una maschera è di norma dotata. Il loro uso rientra fra i prerequisiti minimali dei nostri lettori, pertanto indichiamo solo la specifica proprietà: ***Pulsanti di spostamento***, che per default è = Si. Talvolta potrebbe convenire nasconderli all'utente finale, nel qual caso si imposterà *Pulsati di spostamento = No*. Beninteso, sarà opportuno sostituirli con pulsanti equivalenti più evidenti. Facendo l'esempio di un pulsante di avanzamento di nome, diciamo, *cmdAvanti* anticipiamo che la routine relativa sarà del tipo seguente:

```
Private Sub Avanti_Click()
  DoCmd.GoToRecord , , acNext
End Sub
```

Interruzione di pagina e Struttura a schede

Una maschera molto grande e zeppa di controlli costringe l'utente a tediose esplorazioni con la barre di scorrimento. Per evitarlo la si può suddividere in pagine, inserendo il controllo **Interruzione di pagina**, quartultimo della casella degli strumenti. Basta piazzarlo nel o nei punti opportuni, dopo di che si può navigare da pagina a pagina coi tasti **Pag↓** e **Pag↑**.

Un altro tipo di suddivisione - dell'intera forma ma, volendo anche di una sua parte - si ottiene col controllo **Struttura a schede**, capace di ospitare a sua volta ogni controllo normale associato o meno (caselle, pulsanti ecc.) nelle due o più *schede* di cui è composto. È poi arcinoto che si viaggia da una scheda all'altra con un clic sulla linguetta e nessuno avrà difficoltà ad assegnare le etichette più pregnanti alle varie linguette, per cui aggiungiamo solo, chi non lo sapesse, che più di due schede si ottengono con un clic destro su una linguetta...

Per il momento basta così, ma, a future memoria, riveliamo che pagine e schede fanno parte della libreria di oggetti che compongono una Form.

Il controllo Gruppo di opzioni

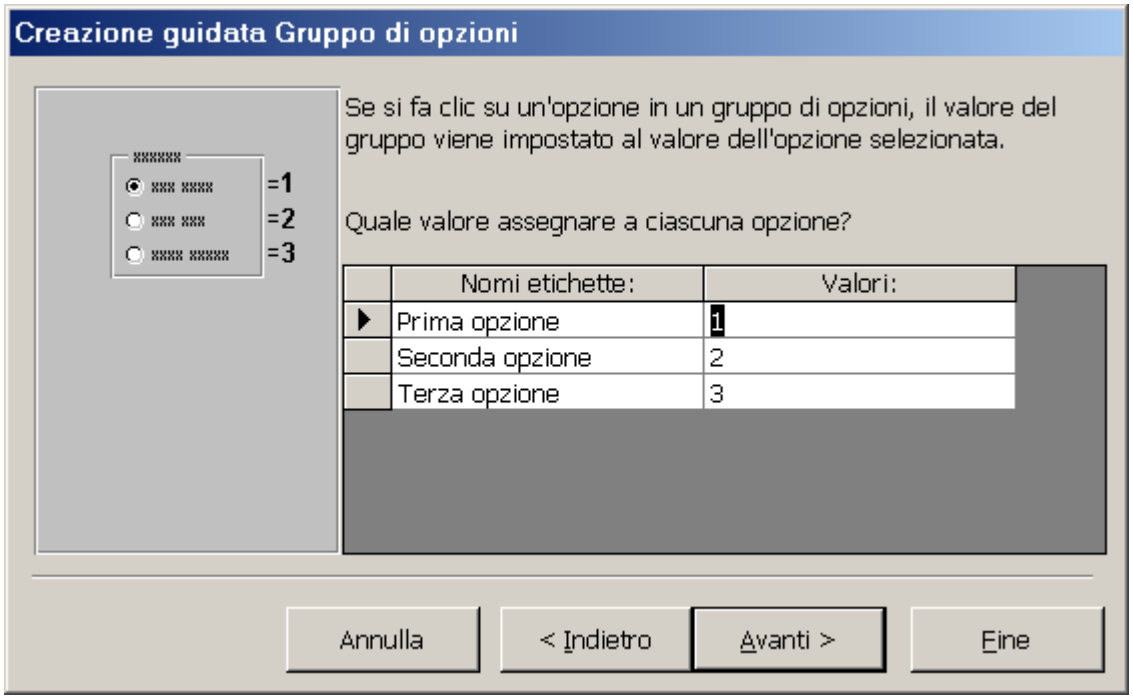
Questo controllo si comporta in modo diverso rispetto al suo parente Excel (là chiamato "Cornice"). La Cornice di Excel è un semplice delimitatore, mentre il controllo **Gruppo di opzioni** di Access VBA racchiude in modo mutuamente esclusivo tre possibili tipi di controllo:

- ***pulsanti di opzione***
- ***interruttori***
- ***caselle di controllo***

Il Gruppo di opzioni va posto sulla maschera *prima* dei controlli da inserire in esso. La cosa più semplice per creare una tale combriccola e comprenderne il funzionamento è il ricorso alla creazione guidata, nel corso della quale ci viene chiesto di scegliere uno dei tre tipi di controlli leciti e il loro numero. Al termine, tutti possono constatare che l'attivazione di uno dei controlli racchiusi disattiva tutti gli altri.

NOTA - Anche le caselle di controllo, generalmente destinate a scelte multiple, poste entro un Gruppo di opzioni si comportano in modo esclusivo, dando luogo a equivoci a nostro avviso da evitare.

La cosa importante da anticipare, per futuri utilizzi VBA, è espressa dal seguente passaggio del wizard:



La finestra mette in evidenza che alle scelte effettuate viene assegnato un preciso valore numerico, da 1 in su. La cosa risulta comodamente sfruttabile, nel codice VBA, con una struttura *Case*:

```
Select Case MieOpzioni
  Case 1
    ... istruzioni su Prima opzione...
  Case 2
    ... istruzioni su Seconda opzione...
  Case 3
    ... istruzioni su Terza opzione...
End Select
```

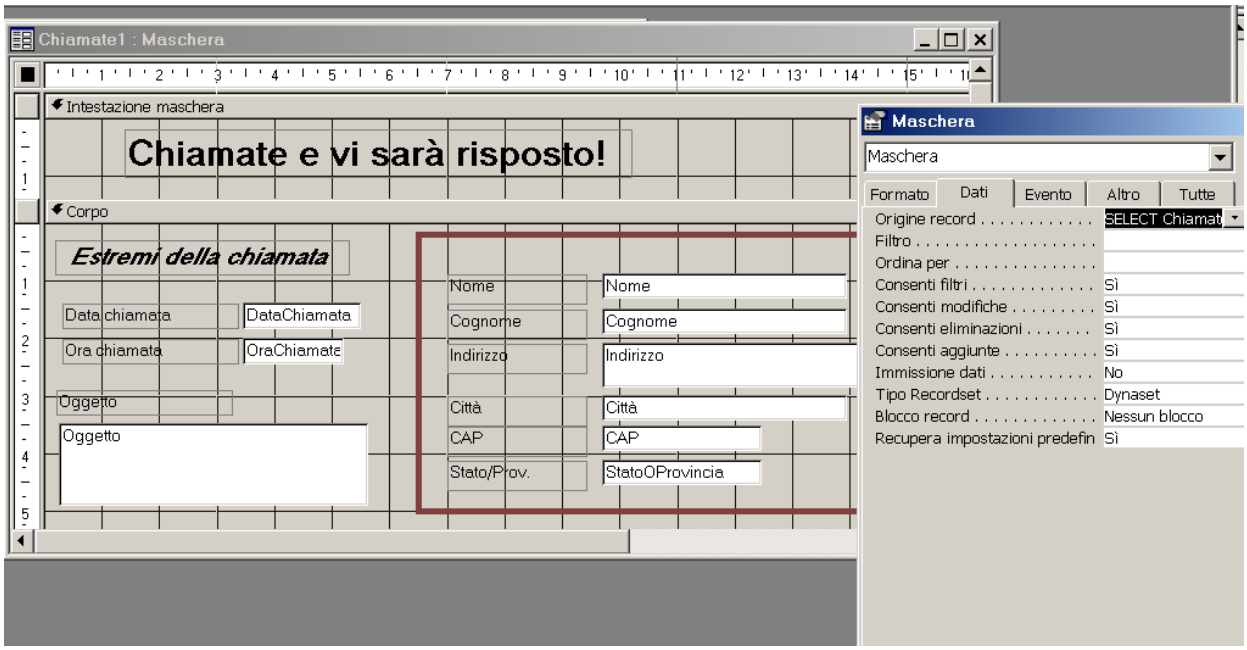
Come meglio si vedrà, *MieOpzioni* è il nome del controllo Gruppo di opzioni ipotizzato dalla precedente figura.

NOTA - Gli utenti di Excel osservino (se non l'hanno già fatto da soli) che questo comportamento, assente coi controlli ActiveX e le Userform, è invece offerto dalla **Casella di gruppo** dei "classici" controlli della barra strumenti Moduli.

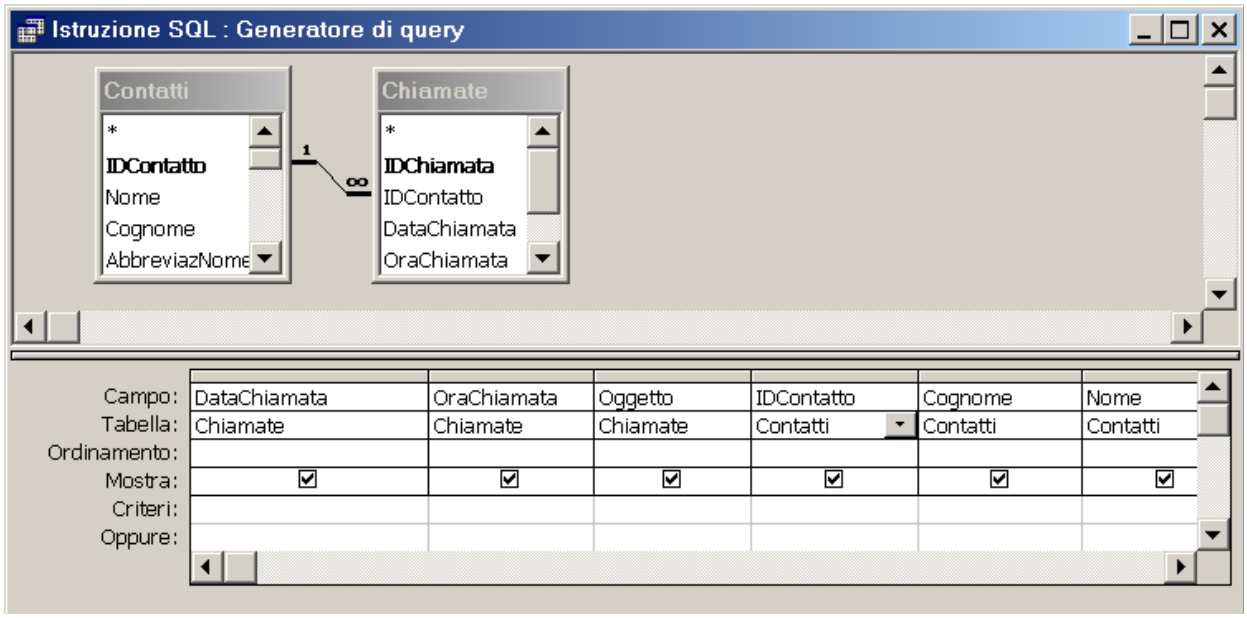
Modificabilità dei dati

Ben più importante dei fattori estetici su cui ci siamo fin qui dilungati è il tema, anzi il problema della modificabilità. Come regola Access - che, lo ripetiamo fino alla noia, opera su disco - permette ogni cambiamento sui campi del record corrente, purché siano rispettate le condizioni di validità fissate in taluni campi (primo esempio che ci viene in mente: sconto non superiore al 10%), salvando la modifica non appena si passa ad altro record, e senza chiedere permesso.

Palesemente ciò è rischioso, e non solo per il tanto bistrattato utente finale ma persino per i più scafati, visto che la distrazione è un male epidemico, che oggi contagia tutti quanti... Per comprendere il problema si consideri la maschera seguente:



Come si intravede nella finestra delle Proprietà la maschera è associata a una query, quella qui sotto riportata (ottenibile con un clic sui Generatori (...) accanto alla casella *Origine record*):



Ci vuol poco a rendersi conto che la query chiama in campo le tabelle Chiamate e Contatti in una evidente relazione Uno-a-molti, in virtù della quale gli estremi di una chiamata (data, ora ecc.) sono associati al soggetto (“contatto”) che l’ha fatta. Ora passi che l’utente possa correggere i primi, ma permettergli di modificare i campi del contatto è il più delle volte inaccettabile. Altri esempi classici si hanno con fatture di vari clienti, animali di vari

proprietari: la modifica “apparentemente” sul lato Molti (in quanto si stanno esaminando dati “di dettaglio” - chiamate, movimenti e simili - si ripercuote (ovviamente) sul lato Uno. In parole povere cambiando il nome del proprietario di un animale lo cambiamo a quello di tutti gli animali che hanno lo stesso padrone, ma operazioni del genere andrebbero fatte, sempre che giuste, sulla tabella dei proprietari, dei contatti ecc. o sulle maschere eventualmente progettate a tale scopo.

NOTA - E se proprio ci fosse un errore di attribuzione? La modifica, se si riflette, va fatta nel Codice identificativo, chiave unica della tabella Contatti (o Proprietari e quant’altro).

Per inibire modifiche si può ricorrere alle seguenti proprietà, le prime dell’oggetto Maschera, l’ultima relativa a controlli:

- ***Tipo recordset***
- ***Consenti modifiche***
- ***Consenti aggiunte***
- ***Consenti eliminazioni***
- ***Immissione dati***
- ***Abilitato***

NOTA - Le proprietà *Blocco record* e *Bloccato* hanno senso in applicazioni multiutente, servono a “bloccare” un intero record oppure un campo, al quale stesse accedendo un altro utente.

Quanto a *Tipo recordset*, premesso che Recordset è un generico oggetto costituito da un insieme di record, come accennato nell’articolo precedente il recordset associato a una query, a una maschera o a un report è un recordset *dinamico*, che cioè è in vita al momento del richiamo della query, maschera o report. E si badi bene che si ha un recordset di tipo dinamico pure quando l’origine della maschera è una semplice tabella.

La proprietà *Tipo recordset* può assumere tre valori: “Dynaset” (default); “Dynaset aggiornamento”; “Snapshot”. Trascurando il secondo caso, impostare la proprietà *Tipo recordset = “Snapshot”* equivale a rendere il recordset di sola lettura.

Le proprietà *Consenti...*, col valore “No” impediscono le modifiche, le aggiunte e le eliminazioni, proprio come il rispettivo nome denuncia, sicché se putacaso *Consenti modifiche = “No”* ma *Consenti aggiunte = “Si”* non si possono modificare i record esistenti ma aggiungerne altri non è vietato.

Quanto a *Immissione dati*, coi valori “Sì” e “No” consente o meno l’input ma questa proprietà, da sola non ha effetto nell’uso manuale. La faccenda sarà chiarita tra poco, esaminando specifiche routine VBA.

Le limitazioni a livello record possono risultare troppo drastiche in certi casi, ed ecco intervenire la proprietà *Abilitato* di valori “Sì” o “No” applicabile ai soli controlli che ci interessano. È il caso da cui siamo partiti (figura precedente) e la cura a questo punto è abbastanza chiara:

1. aprire la maschera in visualizzazione Struttura;
2. selezionare, nella scheda ***Dati*** della finestra Proprietà, le caselle sulla destra (Nome, Cognome, Indirizzo,...);
3. fissare, per tutte, *Abilitato = “No”*.

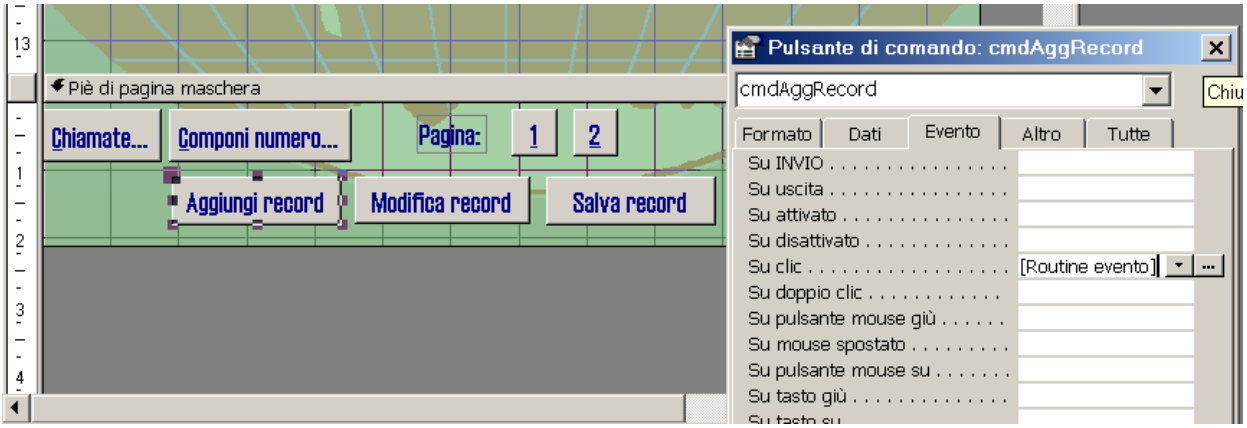
NOTA - Attenzione! La selezione al passo 2 deve interessare le caselle “in quanto tali”. Volendo usare il contornamento con linea il mouse va puntato a partire da un punto delle caselle che escluda le rispettive etichette. Se invece si parte da queste la scheda Dati è vuota...

Maschera di input con cautele VBA

Dopo queste considerazioni sulla modificabilità torna a fagiolo l'esempietto di cui la volta scorsa abbiamo intravisto solo l'inizio, il che ci dà il destro di introdurre, in modo un po' casuale, altri spunti di Visual Basic.

Pulsanti con routine d'evento Clic ottenute con creazione guidata

Scopo di questa semplice esercitazione è di permettere all'utilizzatore di aggiungere, modificare e salvare record a ragion veduta, la qual cosa verrà realizzata attivando espliciti pulsanti di comando. Ci serviremo della maschera **Contatti** del database **Gestione contatti1** facilmente ottenuto col generatore di database (v. articolo precedente). In fondo a tale maschera, nel suo piè di pagina (*) aggiungiamo tre pulsanti di comando, più o meno come quelli illustrati nella figura seguente:

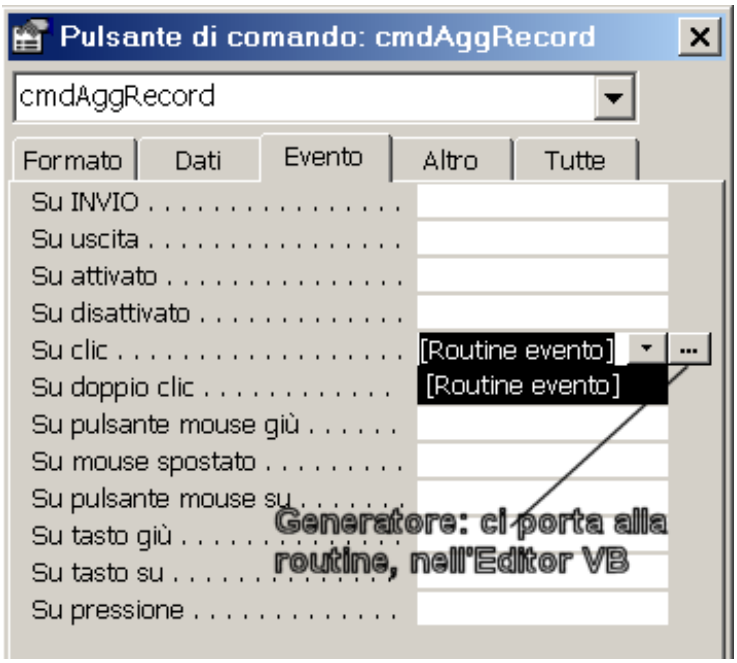


NOTA (*) - L'inserimento di controlli che lanciano routine nel piè di pagina è importante ogni volta che la maschera è suddivisa in pagine, come in questo caso, o in schede (usando il controllo **Struttura a schede**, v.).

Alla base di questa elementare esercitazione c'è anche l'idea di rivolgersi a un utente finale che non conosce certe particolarità di Access, come l'aggiunta di un record che in verità si può fare con un clic sull'apposito pulsante di spostamento standard, quello segnato con * (asterisco).

Ma ecco il dettaglio delle operazioni da compiersi.

1. Nella vista Struttura impostare la proprietà *Consenti modifiche = "No"* dell'oggetto Maschera.
2. Assicurarsi che sia attivo il pulsante **Creazioni guidate controllo** (bacchetta magica) nella casella degli strumenti, poi scegliere il pulsante di comando e posizionarlo nel piè di pagina maschera.
3. Seguire i vari passaggi del wizard, scegliendo l'azione "Aggiungi nuovo record" della categoria "Operazioni su record", assegnando l'etichetta "Aggiungi record" e il nome *cmdAggRecord* al neonato pulsante.
4. Per farla breve, al termine possiamo recarci nell'Editor VBA con Alt+F11 o, in modo più focalizzato, attivare i tre puntini del generatore accanto alla proprietà *Su clic* del pulsante:



La dicitura "[Routine d'evento]" è una spia della presenza della Sub che stiamo per vedere, anzi rivedere:

```
Private Sub cmdAggRecord_Click()
On Error GoTo Err_cmdAggRecord_Click

  DoCmd.GoToRecord , , acNewRec
Exit_cmdAggRecord_Click:
  Exit Sub

Err_cmdAggRecord_Click:
  MsgBox Err.Description
  Resume Exit_cmdAggRecord_Click
End Sub
```

Questa come le altre routine d'evento dei controlli di una maschera si trovano nel *modulo di classe* **Form_Contatti** nell'Editor VBA, in modo del tutto analogo a quanto accade con le Form del VB6 e le UserForm di Excel. Un modulo di classe, che corrisponde alle Form Visual Basic e alle UserForm di Excel o Word, viene anche chiamato *modulo di maschera*.

NOTA - Altri moduli di classe si hanno con gli oggetti Report, come si vedrà.

Il canovaccio *On Error* con le due etichette *Exit_cmdAggRecord_Click* e *Err_cmdAggRecord_Click* ecc. serve a una gestione di eventuali errori. Prendiamo piuttosto nota del codice operativo:

```
DoCmd.GoToRecord , , acNewRec
```

5. Ripetere i passi da 3 a 4 per il pulsante **Salva record**, mutatis mutandis, ossia scegliendo l'azione "Salva record" nella categoria "Operazioni su record". La macro risultante, in cui abbiamo evidenziato in grassetto il codice di salvataggio e una piccola aggiunta nella gestione errori, è questa:

```
Private Sub cmdSalvaRecord_Click()
On Error GoTo Err_cmdSalvaRecord_Click

  DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
Exit_cmdSalvaRecord_Click:
  Exit Sub

Err_cmdSalvaRecord_Click:
  MsgBox "Non posso salvare una maschera vuota!"
  Resume Exit_cmdSalvaRecord_Click
End Sub
```

La "piccola aggiunta" è consistita semplicemente nel rendere più chiaro il messaggio standard *Err.Description* (che pure Access emette in italiano) proveniente dall'errore che si ha quando l'utente pretendesse di salvare un record che non ha riempito affatto.

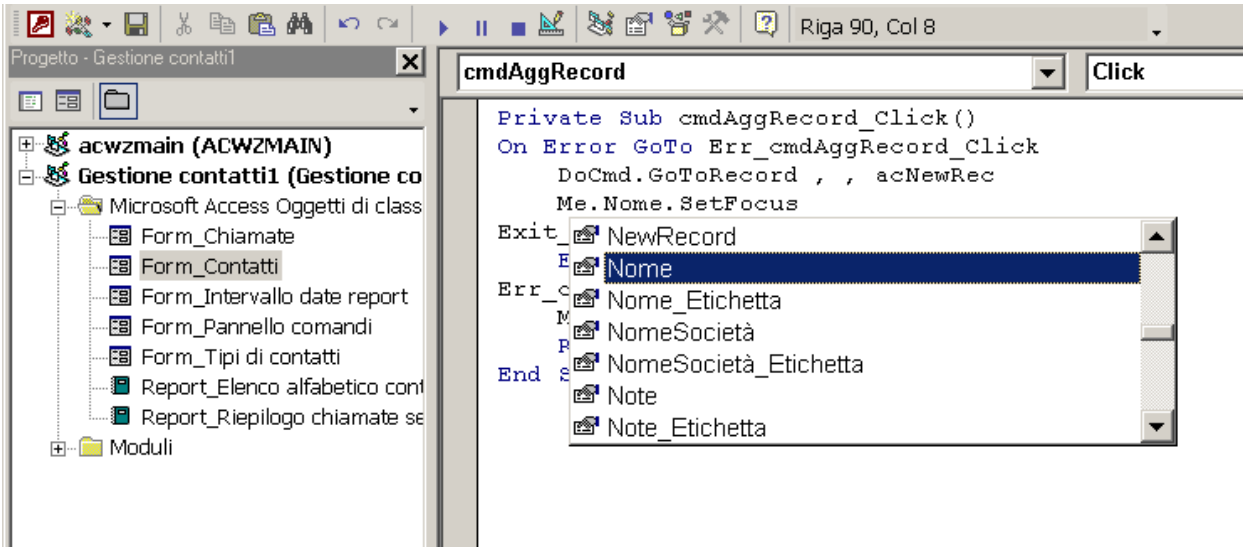
Per il momento ci accontentiamo di ripetere che ***DoCmd*** ("esegui comando") è un oggetto dotato di parecchi metodi, ciascuno con vari argomenti, taluni dei quali si possono omettere, mantenendo però le virgole. Col metodo *GoToRecord* è chiaro il significato dell'argomento *acNewRec*, mentre il comando di salvataggio è decisamente più criptico. Si intuisce che *DoMenuItem* chiama in gioco un comando di menu della barra della maschera (argomento *acFormBar*) con argomento *acSaveRecord...* Fermiamoci qui, perché una trattazione più completa di *DoCmd* richiede quasi un intero articolo, che verrà fuori in seguito.

Modifichiamo invece la prima routine come segue, depurandola dal codice di gestione errori:

```
Private Sub cmdAggRecord_Click()
    DoCmd.GoToRecord , , acNewRec
    Me.Nome.SetFocus
End Sub
```

Il codice *Me.Nome.SetFocus* applica il metodo *SetFocus* alla casella *Nome*. Il "focus" posto sul primo campo della maschera lo predispone alla digitazione dell'utente. Andava anche bene *Nome.SetFocus* in quanto *Me*, sinonimo della Form corrente, si può omettere.

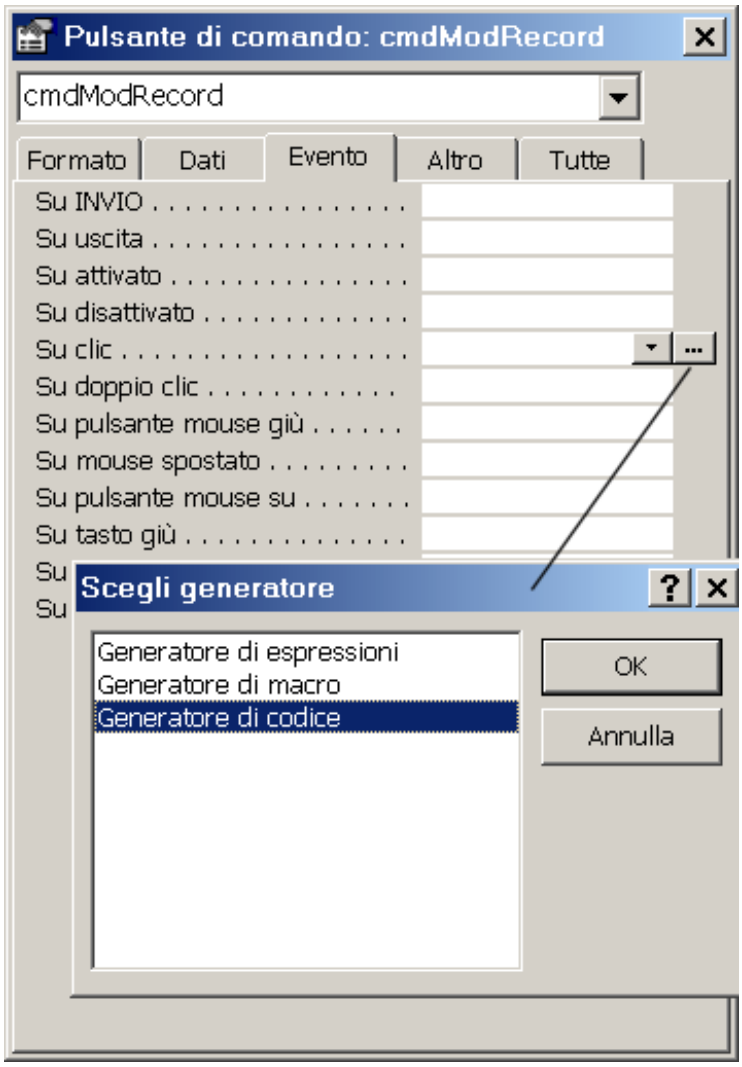
C'è un piccolo vantaggio a usare *Me*. Se lo si fa seguire dal punto (.) si spalanca accanto ad esso la lista dei controlli, metodi e proprietà di cui l'oggetto Form corrente è padre:



Pulsanti con routine d'evento Clic senza creazione guidata

Proseguiamo con la creazione del pulsante **Modifica record**. Stavolta è giocoforza rinunciare alla creazione guidata, perché il wizard non offre nessuna azione del genere nella categoria "Operazioni su record".

6. Disattivare il pulsante del wizard, scegliere l'icona pulsante di comando e posizionarlo nel piè di pagina maschera come già detto.
7. Nella finestra Proprietà del nuovo controllo assegnare il *Nome = "cmdModRecord"* e la *Etichetta = "Modifica Record".*
8. Nella scheda Eventi attivare la casella *Su clic* e schiacciare i tre puntini, poi nella susseguente finestrella scegliere **Generatore di codice**.



Access ci porta nell'Editor VBA, in corrispondenza della specifica routine d'evento, stavolta limitata ai suoi estremi:

```
Private Sub cmdModRecord_Click()

End Sub
```

A noi il compito di riempirla. Dopo ricerche affannose e qualche mal di testa si scopre che la proprietà che fa per noi è *AllowEdits*, ed ecco il completamento dovuto:

```
Private Sub cmdModRecord_Click()
  Me.AllowEdits = True
End Sub
```

Ma noi vogliamo che questo permesso scada appena si passa a un altro record (semplicemente con le freccette di navigazione di cui una maschera è normalmente dotata) e, a tal fine, ricorriamo all'evento **Su corrente**, che si scatena appunto in tale circostanza. Se poi si riflette, la stessa disabilitazione potrebbe essere necessaria anche dopo che il record è stato aggiornato. Con le mosse già viste, le due routine d'evento sono presto create:

```
Private Sub Form_Current()
  Me.AllowEdits = False
End Sub

Private Sub Form_AfterUpdate()
  Me.AllowEdits = False
End Sub
```

Lo spazio ci costringe a fermarci qui. Al lettore proponiamo due domande: a) entrambe le routine precedenti sono indispensabili?; b) è opportuno o non aggiungere *Me.AllowEdits = True* alla Sub di salvataggio?

Una cosa ci preme porre in risalto, anzi due:

a) contrariamente a quel che si potrebbe credere, la proprietà *AllowEdits* equivale a *Immissione dati* dell'uso manuale e non a *Consenti modifiche*, che *non* ha corrispettivi in VBA;

b) mentre *Immissione dati*, come s'è visto, è inoperante in ambito manuale, *AllowEdits* agisce sì in ambiente programmatico ma solo a condizione che sia fissato *Consenti Modifiche = "No"* nel progetto della maschera.

Provare per credere. In particolare, con una maschera normale, priva di ogni divieto, ci si porti nell'Editor VBA, poi nella finestra Immediata (attivarla con **Ctrl**+**G**) e si digiti il comando VB *Me.AllowEdits = False*. Al ritorno nella maschera si constaterà che solo la proprietà *Immissione dati* è stata fissata a "No".

Routine d'evento tramite funzione su modulo standard

Completiamo questa rapida escursione in VBA accennando a una particolarità di Access:

Le routine d'evento si possono abbinare anche a funzioni normali, su un modulo standard.

Per fissare l'idea riferiamoci all'evento Clic del nostro pulsante **Aggiungi record**. Ecco i passi da compiersi, in modalità Struttura:

1. Per prudenza, eliminare la routine *cmdAggRecord_Click* nel modulo di classe **Form_Contatti**.
2. Passare all'Editor VBA con **Alt**+**F11** (o con un clic sull'icona "Codice" della barra standard) e creare un nuovo **Modulo1**, di tipo generale, e digitare in esso questa routine:

```
Function AggiungiRecord()
  DoCmd.GoToRecord , , acNewRec
End Function
```

3. Tornare con **Alt**+**F11** (o **Alt**+**Q**) alla maschera Contatti e, nella casella dell'evento Clic del pulsante in questione, scrivere, semplicemente: "=AggiungiRecord()".

NOTA - La digitazione va fatta a mano, non serve il pur versatile Generatore di espressioni, che ignora le funzioni create da noi (contrariamente a quanto accade in Excel, in cui peraltro un Generatore non c'è).

Tornando nella vista Maschera si constaterà lo stesso effetto di prima.

Questa faccenda sorprenderà molti utenti di Excel che non pensano a una funzione capace di compiere operazioni! Per convincersi che anche in Excel ciò non è vietato si provi, in ambiente Excel, il codice seguente:

```
Function MiaStranaFunz()
  With ActiveCell
   .Copy .Offset(0, 1)
   'copia a destra la cella corrente
  End With
End Function

Sub ProvaMiaStranaFunz
  x = MiaStranaFunz
End Sub
```

È evidente che in Excel la cosa è poco più che una curiosità. Ma si può dire lo stesso in Access? Quel che è certo è che si tratta di un residuo di edizioni passate, quando esistevano solo moduli standard e le routine d'evento predefinite racchiuse in moduli di oggetti Form sono decisamente più organiche e "moderno". Tuttavia il vecchio stile due vantaggi li ha, specie in assenza di argomenti predefiniti:

a) una funzione come la nostra *AggiungiRecord* (o molto più complessa) opera con qualsiasi maschera, non va riscritta tutte le volte;

b) copiando un controllo da una maschera a un'altra maschera (cosa che può far risparmiare tempo) non si deve riscrivere la sua routine d'evento nella nuova maschera (cosa che può far perdere tempo).

Il dibattito è aperto, comunque terminiamo facendo notare che una funzione del genere può anche esser dotata di argomenti. Esempio, di discutibile umoristico:

```
Function MiaBuffaFunz(Segreto As Boolean, Mess As String)
  If Not Segreto Then MsgBox "Fai sapere a tutti che " & Mess
  If Segreto Then MsgBox "Non dire in giro che " & Mess
End Function
```

La vita offre altri divertimenti, ma in mancanza di meglio il lettore può provare a inserire nella proprietà Clic di un pulsante le due alternative seguenti:

=MiaBuffaFunz(Vero;"Il cacio è buono con le pere")

=MiaBuffaFunz(Falso;"Il cacio è buono con le pere")

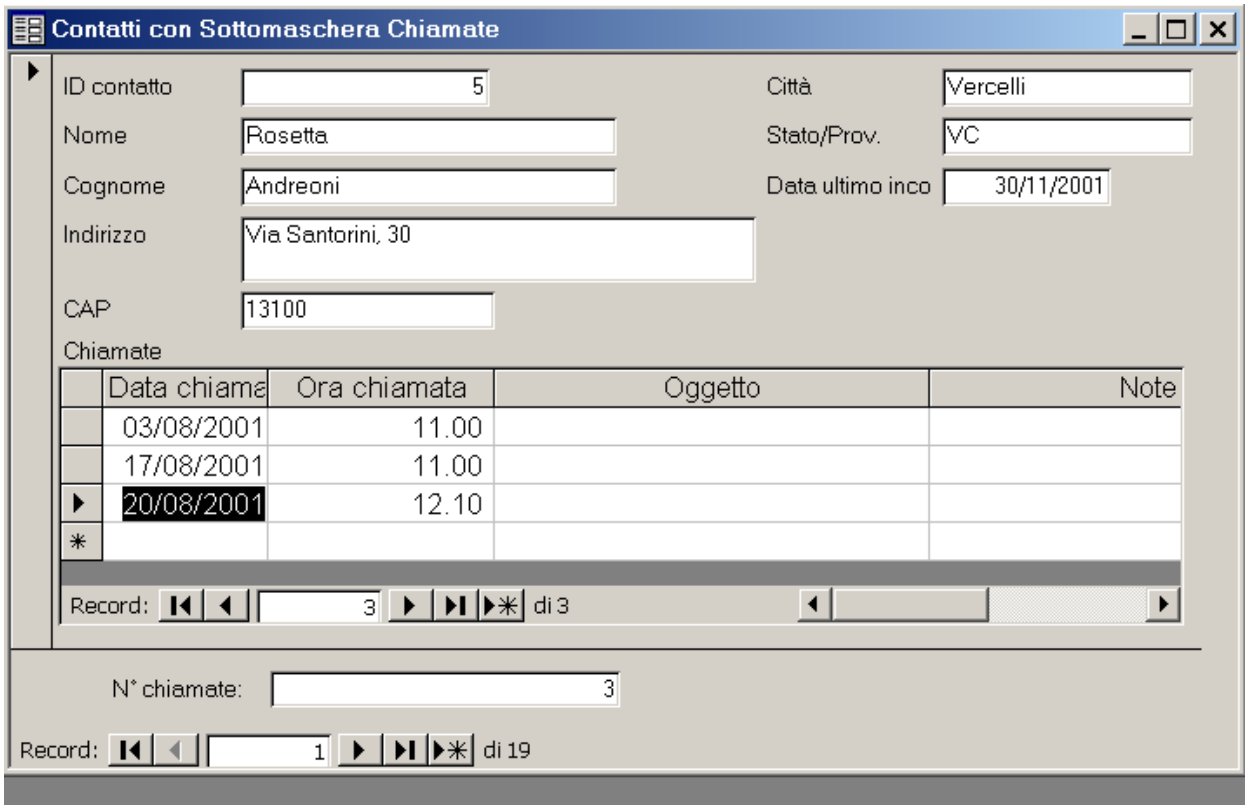
A proposito: come si nota *la sintassi delle funzioni richiamate nelle maschere e report, nell'edizione italiana di Access, è italiana*; *in particolare* True *e* False *vanno espressi con* Vero *e* Falso, *il separatore di argomenti è il punto e virgola e la virgola separa i decimali*, mentre l'ambiente VBA resta inglese.

NOTA - La situazione è insomma del tutto simile a quella del mondo Excel.

Maschere, sottomaschere e campi di totale

Una qualità molto importante è la possibilità di collegare gerarchicamente due maschere, in modo che la maschera principale o maschera-madre richiami una sottomaschera o maschera-figlia. Ciò è particolarmente utile in abbinamento con le query Uno-a-molti, in cui il lato Uno fornisce i dati della maschera principale mentre il lato molti fornisce quelli della sottomaschera.

Per entrare in medias res, riferiamoci all'esempio tipico della figura seguente:



La sottomaschera ha nome **Sottomaschera Chiamate** (come vedremo, ha vita a sé) è incorporata nella parte inferiore della maschera principale, di nome **Contatti con sottomaschera chiamate**. Tale sottomaschera, in forma di foglio dati (di solito il più conveniente con le sottomaschere), le chiamate di pertinenza di ciascun Contatto.

Non c'è quasi bisogno di aggiungere che è la solita relazione Uno-a-molti su cui si fonda il nostro database, già richiamata in una figura precedente di questo stesso articolo. Piuttosto vale la pena di ricordare che un'evidenziazione più semplice ma concettualmente identica

Access 2000 / 2002 la offre pure a livello tabelle, basta aprire la tabella Contatti e cliccare su una (o più) delle croci nella prima colonna per rendersene conto:

Contatti : Tabella

ID contatto	Nome	Cognome	AbbreviazNome	Indirizzo	Città	Stato/Prov.
5	Rosetta	Andreoni	NDRRST	Via Santorini, 30	Vercelli	VC
4	Carmelo	Cacace	CCCCRM	Viale Kennedy, 89	Novara	NO
9	Toni	Contarini	CNTTNI	Piazza Garibaldi, 1	Novara	NO
18	Gennarino	Esposito	SPSGNN	Via Dante, 45	Novara	NO
11	Maria	Gariboldi	GRBMRI	Via Fiori Oscuri, 12	Milano	MI
14	Camillo	Giannattasi	GNNCML	Piazza Minerva, 11	Pavia	PV
6	Smitherson	John	SMTJHN	8 Wilbure Street	Swindon	Wiltshire
8	Pasquale	Landolfi	LNDPSQ	Piazza Garibaldi, 1	Novara	NO
15	Genoveffa	Lanfranchi	LNFGNV	Via Leopardi, 32	Pavia	PV
3	Giovanni	Mastropasqua	MSTGVN	Via Sartorini, 30	Vercelli	VC
12	Andrea	Mazzarini	MZZNDR	Via Fiori Oscuri, 12	Milano	MI
13	Rosa	Mazzarini	MZZRSA	Via Fiori Oscuri, 12	Milano	MI
7	Carmen	Pieri	PRICRM	Piazza Garibaldi, 1	Novara	NO
2	Maria	Pintacuda	PNTMRI	Piazza Oberdan, 3	Milano	MI
17	Renato	Pistolesi	PSTRNT	Via Roma, 40	Vercelli	VC

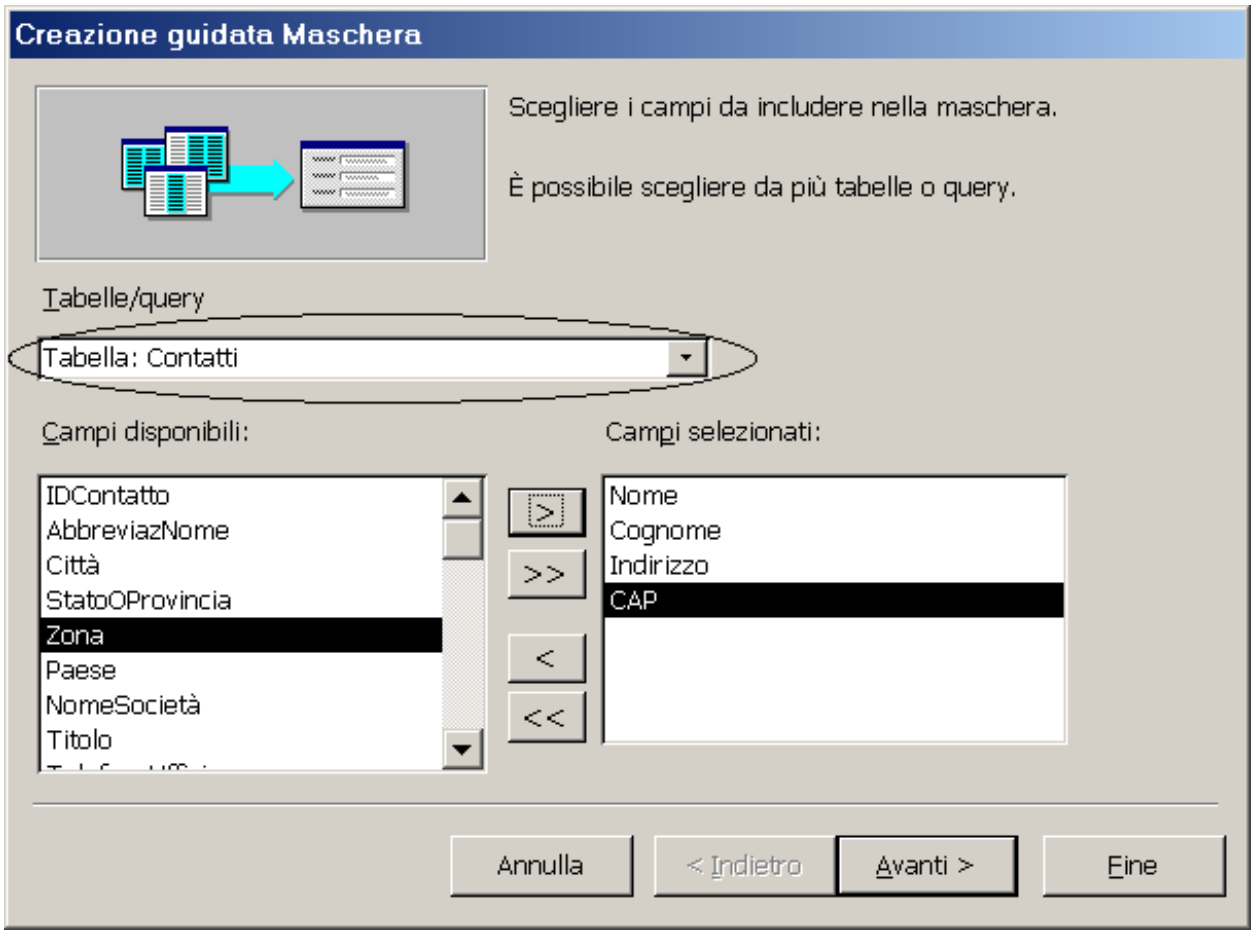
ID chiamata	Data chiamata	Ora chiamata	Oggetto	Note
3	03/08/2001	11.00		
4	17/08/2001	11.00		
10	20/08/2001	12.10		
(Contatore)				

ID chiamata	Data chiamata	Ora chiamata	Oggetto	Note
13	25/08/2001	10.00		
23	30/10/2001	12.00		
(Contatore)				

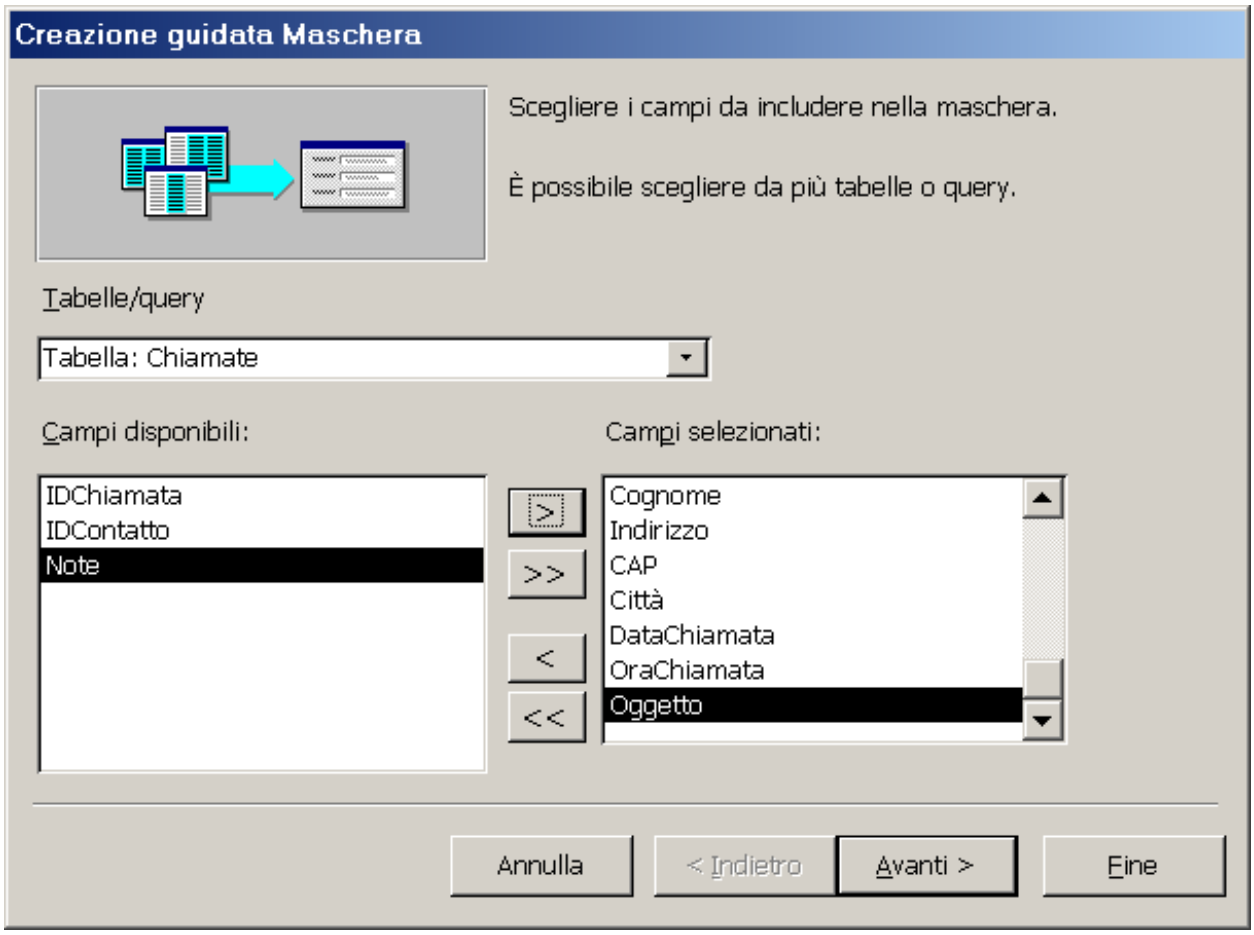
Creazione guidata di una maschera con sottomaschera

Per ottenere il risultato di cui alla penultima figura, si compiono le mosse seguenti.

1. Avviare la creazione guidata di maschere con l'apposito pulsante dell'ambiente Maschere della finestra Database.
2. Al primo passaggio indicare la prima tabella - **Contatti** nel nostro caso - poi spostare i campi scelti con le facili mosse qui sotto illustrate:



3. **Punto cruciale!** Non cliccare su **Avanti >** (come si fa con le maschere semplici) bensì scegliere la seconda tabella nel combobox *Tabelle/query* (evidenziato con un'ovale nella figura precedente). Nel nostro caso scegliere **Chiamate**, poi accodare i campi di questa ai precedenti:



4. Un clic su **Avanti >** presenta ora una finestra (che non appare quando si seleziona una sola tabella!), nella quale sono possibili due scelte: a) **In base a contatti** vs. **In base a chiamate**; b) **Maschera con sottomaschera** vs. **Maschere collegate**. La prima alternativa definisce la maschera rispetto alla sottomaschera, pertanto noi scegliamo a). Pure sulla seconda alternativa non dovremmo aver dubbi su a).
5. Un ulteriore quadro ci offre le opzioni **Tabulare**, **Foglio dati** (default), **Tabella pivot** e **Grafico pivot**. Per quanto detto il default è di solito preferibile.
6. All'ultimo passaggio il wizard suggerisce un nome tipo "Contatti" o "Contatti1", "Contatti2"... per la maschera e un nome "Chiamate sottomaschera" per la maschera-figlia. Noi abbiamo cambiato il primo in "Contatti con sottomaschera chiamate" e il secondo in "Sottomaschera Chiamate". De gustibus...

Già che ci siamo, precisiamo che la scelta **Maschere collegate** (che ognuno può facilmente provare da sé) dà origine a due maschere separate ma unite da legame... filiale. La madre reca un apposito pulsante, cliccando sul quale viene aperta la figlia. Ne riparleremo in futuro? Solo se se ne presenterà l'occasione.

Creazione manuale di una maschera con sottomaschera

Questo metodo più rapido e, a modo suo, brillante potrebbe tornare utile quando due maschere esistono già. Per fissare le idee si abbiano due maschere, diciamo **Contatti con Chiamate** e **Sottomodulo Chiamate**. Possiamo pensare di crearle entrambe ex novo separatamente, con la creazione guidata (saltando il cruciale punto 3) oppure salvando sotto tali nomi maschere preesistenti e modificandole secondo le proprie esigenze.

A questo punto, ecco il da farsi.

1. Aprire la maschera **Contatti con chiamate**, in visualizzazione Struttura.

2. Assicurandosi che sia aperta la finestra Database nella sezione Maschere, selezionare l'icona **Sottomodulo chiamate** (n.b. *senza* aprire questa candidata-sottomaschera).
3. Trascinare tale icona, rilasciandola nella zona di **Contatti con chiamate** destinata a ospitare la sottomaschera.

Se si è avuta la necessaria pazienza, il risultato sarà lo stesso ottenuto col metodo precedente.

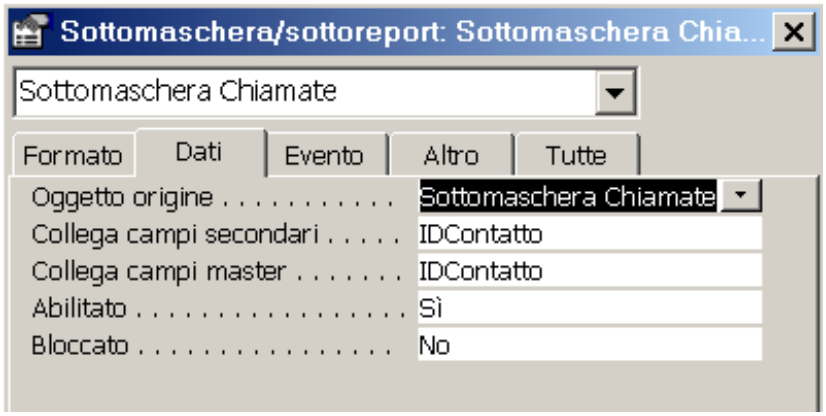
NOTA - È infine importate sottolineare che, comunque create, le maschere con sottomaschere o le maschere collegate presuppongono una *relazione*, fissata a livello database.

Caratteristiche delle maschere con sottomaschere

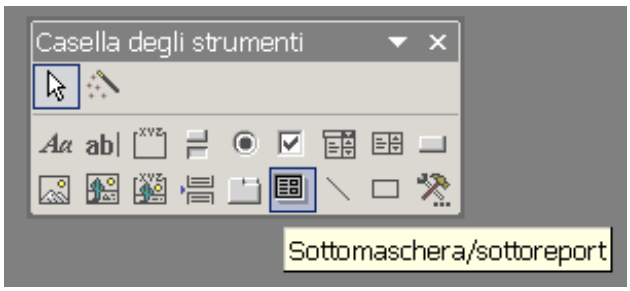
Se andiamo a esaminare la form **Sottomaschera Chiamate** creata col wizard avremo una sorpresa (almeno la prima volta): essa offre la vista maschera, non quella foglio dati. O, perlomeno, si può passare alla vista maschera senza problemi. Il contrario non è possibile se si attiva in vista Struttura la maschera principale e, su questa, si seleziona la sottomaschera. Qualcuno sa dire perché? La risposta è in fondo al paragrafo.

Ma ecco i punti strutturali basilari.

- La sottomaschera presenta le proprietà *Collega campi secondari* e *Collega campi master* impostate (com'era da attendersi) rispettivamente sui campi che identificano la maschera-figlia e su quelli della maschera-madre (o *master*). Si tratta, palesemente, dei campi della relazione Uno-a-molti su cui si fonda il legame "parentale" maschera/sottomaschera, costituito da IDContatto nel nostro caso (ma l'omonimia nelle join non è obbligatoria, si ricordi).



- Quando la sottomaschera ha il layout foglio dati essa è racchiusa in un particolare controllo, denominato **Sottomaschera/sottoreport** (da usare pure coi report padri/figli, si intuisce):



- Se invece la sottomaschera è stata creata col layout tabulare (sconsigliato, sempre a parer nostro) essa risulta in certo senso incorporata nella maschera principale.

Il secondo punto suggerisce subito una *terza* possibilità di creazione di una sottomaschera, con le grandi mosse seguenti: 1. creare a parte la sottomaschera; 2. selezionare il controllo

Sottomaschera/sottoreport e posizionarlo nell'area opportuna della maschera mamma; 3. fissare opportunamente le proprietà *Oggetto origine* e le due di collegamento testé viste.

Ma forse bastano e avanzano i primi due metodi e più bravi & volonterosi sono a questo punto invitati a sperimentare per proprio conto questo terzo metodo.

Risposta al quesito

La descrizione appena fatta implica la risposta: non va fatta confusione fra visualizzazione e layout, quest'ultimo non può essere fatto con un clic, ma ridisponendo i controlli (rivedere se occorre il paragrafo "Le viste- anzi layout! - tabulare e giustificata" ← *cliccare qui per rileggerlo*).

Introduzione alle funzioni di totalizzazione

Se si rivede la figura subito sotto al paragrafo Maschere, sottomaschere e campi di totale (← *cliccare qui per rileggerlo*) si noterà la presenza nella parte inferiore della sottomaschera un campo etichettato con "Numero chiamate:". Scorrendo i vari record (della maschera) varia il numero di chiamate di pertinenza di questo o quel contatto, il che ci sembra corretto. Ma vediamo cosa c'è sotto.

Per creare tale casella di testo si debbono compiere operazioni piuttosto complesse (e delicate!). Vediamole.

1. Aprire la form **Sottomaschera Chiamate** in visualizzazione Struttura, vista che permette di evidenziarne il piè di pagina (che la vista Foglio dati occulta).
2. Allargato opportunamente tale piè di pagina, inserirvi una casella di testo (non associata), assegnandole il nome **NumChiamate** e, alla relativa etichetta, il valore "Numero chiamate:".
3. Aperta se occorre la finestra Proprietà assegnare a *Origine controllo* la formula seguente:

```
=Conteggio([IDContatto])
```

4. Salvando e riaprendo la sottomaschera, in vista Maschera, si constaterà che nella nuova casella vengono conteggiati tutti i record della tabella Chiamate.
5. Si apra ora, in vista Struttura, la mamma **Contatti con sottomaschera chiamate** e ripetere nel piè di pagina di quest'ultima le operazioni 1 e 2.

NOTA - Dare lo stesso nome - NumChiamate - a caselle di maschere diverse è più che lecito, anzi a volte opportuno.

6. assegnare alla proprietà *Origine controllo* la formula seguente (da scrivere tutta di seguito):

```
=[Maschere]![Contatti con sottomaschera Chiamate].[Sottomaschera Chiamate].[Form]!
[NumChiamate]
```

Si può ora constatare che la conta è, per ogni contatto, limitato alle chiamate relative. Bello, ma di sicuro nella mente dei nostri (pazienti) interlocutori si affolla una serie di domande... serie. In primo luogo appare, se non chiaro, senz'altro intuitivo che l'ultima formulaccia richiama il campo **NumChiamate** della maschera-figlia. Ma qual è la sintassi? Le persone più spericolate proveranno poi a fare a meno del campo di conteggio della sottomaschera provando in vari modi ad applicare la funzione *Conteggio* alla sottomaschera dall'interno della maschera master. Senza risultato. Ma perché?

Le funzioni di totalizzazione, come ***Conteggio*** (inglese *Count*) o ***Somma*** (inglese *Sum*), sono della massima importanza nelle maschere, nelle form (ancor più) e persino nelle query. L'argomento è ghiotto, perché implica un'idea di programmazione senza accorgersene, analoga alle formule di Excel, ma con particolarità dovute al diverso ambiente Access.

Ne riparleremo in abbondanza, ma questo è un fogliettone (feuilleton) e la puntata termina qui.